Available online at www.sciencedirect.com
ScienceDirect
European Journal of Operational Research 175 (2006) 246–257
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/ejor

Discrete Optimization

A new approach to solving the multiple traveling salesperson problem using genetic algorithms

Arthur E. Carter [a,1], Cliff T. Ragsdale [b,*]

[a] College of Information Science and Technology, Radford University, P.O. Box 6933, Radford, VA 24142, United States
[b] Department of Business Information Technology, Pamplin College of Business, 1007 Pamplin Hall, Virginia Tech, Blacksburg, VA 24061, United States

Received 20 October 2003; accepted 14 April 2005
Available online 27 June 2005

Abstract

The multiple traveling salesperson problem (MTSP) involves scheduling $m > 1$ salespersons to visit a set of $n > m$ locations so that each location is visited exactly once while minimizing the total (or maximum) distance traveled by the salespersons. The MTSP is similar to the notoriously difficult traveling salesperson problem (TSP) with the added complication that each location may be visited by any one of the salespersons. Previous studies investigated solving the MTSP with genetic algorithms (GAs) using standard TSP chromosomes and operators. This paper proposes a new GA chromosome and related operators for the MTSP and compares the theoretical properties and computational performance of the proposed technique to previous work. Computational testing shows the new approach results in a smaller search space and, in many cases, produces better solutions than previous techniques.

Keywords: Genetic algorithms; Scheduling; Traveling salesman

1. Introduction

The multiple traveling salesperson problem (MTSP) can be used to model many practical problems. The MTSP is similar to the notoriously difficult traveling salesperson problem (TSP) that seeks an optimal tour of n cities, visiting each city exactly once with no sub-tours. In the MTSP, the n cities must be partitioned into m tours, with each tour resulting in a TSP for one salesperson. The MTSP is more difficult than the TSP because it requires determining which cites to assign to each salesperson, as well as the optimal ordering of the cities within each salesperson's tour.

* Corresponding author. Tel.: +1 540 231 4697.
E-mail addresses: aecarter@radford.edu (A.E. Carter), crags@vt.edu (C.T. Ragsdale).
[1] Tel.: +1 540 831 5238.

0377-2217/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2005.04.027

Perhaps the most common application of the MTSP is in the area of scheduling. The scheduling of jobs on a production line is often modeled as a TSP. If the production operation is expanded to have multiple parallel lines to which the jobs can be assigned, the problem can be modeled as a MTSP (Carter and Ragsdale, 2002). Another problem that is often modeled as a MTSP is the vehicle scheduling problem (VSP). The VSP consists of scheduling a set of vehicles, all leaving from and returning to a common depot, to visit a number of locations such that each location is visited exactly once (Park, 2001). A variation on the TSP that can also be modeled as a MTSP involves using one salesperson to visit n cities in a series of m smaller tours. This describes the scheduling problem of sales/service personnel that visit n cities over a period of time but travel during the week and return home on weekends.

Due to the combinatorial complexity of the MTSP, it is necessary to employ heuristics to solve problems of realistic size. Genetic algorithms (GAs) represent one type of heuristic that researchers have applied to the TSP. More recently, researchers studying the VSP have expanded the use of GAs developed for the TSP to also address the MTSP. Most of the research on using GAs for the VSP has focused on using two different chromosome designs for the MTSP. Both of these chromosome designs can be manipulated using classic GA operators developed for the TSP; however, they are also prone to produce redundant solutions to the problem. This research introduces a new chromosome for the MTSP that works with classic GA TSP operators while dramatically reducing the number of redundant solutions in the solution space, thereby improving the efficiency of the search.

The remainder of this paper is organized as follows. First, we review the literature on solving the TSP and MTSP using GAs. Next, we provide an overview of GAs focused on the characteristics associated with well-designed chromosomes. Next, the two chromosomes traditionally used for the MTSP are presented and contrasted with our new proposed chromosome. Finally, a computational comparison of the various solution approaches is presented followed by concluding remarks and suggestions for future research in this area.

2. Literature review

A number of different methods have been proposed for obtaining either optimal or near optimal solutions for the TSP. The methods used to solve the TSP range from classic methodologies based on linear programming (Wong, 1980) and branch-and-bound (Little et al., 1963; Bellmore and Malone, 1971) to artificial intelligence methods such as neural networks (Shirrish et al., 1993), tabu search (Glover, 1990), and GAs (Goldberg and Lingle, 1985). For a good overview of the TSP and various proposed solutions methodologies for this problem see Lawler et al. (1985).

Given the combinatorial complexity of TSPs of realistic size, a solution methodology that efficiently solves TSPs to global optimality remains elusive. While advances have been made in solving the TSP, those advances have come at the expense of increasingly complex computer code (Chatterjee et al., 1996).

Most of the work on solving MTSPs using GAs has focused on the VSP (Malmborg, 1996; Park, 2001). The VSP consists of scheduling a fleet of m vehicles to visit n cities with each city being visited by one and only one vehicle. The VSP typically includes constraints on the number of cities each vehicle can visit due to the capacity of each vehicle and the size of the load to be picked up at each city. In some cases, the cities must be visited within specific time windows. These issues lead to a number of different possible configurations for the VSP: those with or without time windows, those with heterogeneous or homogeneous vehicle capacities, and those with travel distance and/or fleet size restrictions. A variety of objectives can also be considered, including: minimize the total (or maximum) distance, minimize the number of vehicles required, and minimize the number of time window violations.

3. Overview of GAs

Genetic algorithms (GAs) are a relatively new optimization technique that can be applied to TSPs. The basic ideas behind GAs evolved in the mind of John Holland at the University of Michigan in the early 1970s (Holland, 1975). GAs were not originally intended for highly constrained optimization problems but were soon adapted to order-based problems like the TSP (Goldberg and Lingle, 1985; Jog et al., 1989; Whitley et al., 1989). The development of effective GA operators for TSPs led to a great deal of interest and research to improve the performance of GAs for this type of problem (Poon and Carter, 1995; Qu and Sun, 1999; Katayama et al., 2000). Several summaries of solving TSPs with GAs have been published that provide comprehensive reviews of the operators and associated issues (Potvin, 1996; Schmitt and Amini, 1998; Schaffer et al., 1991).

In a nutshell, GAs work by generating a population of numeric vectors (called chromosomes), each representing a possible solution to a problem. The individual components (numeric values) within a chromosome are called genes. New chromosomes are created by crossover (the probabilistic exchange of values between vectors) or mutation (the random replacement of values in a vector). Mutation provides randomness within the chromosomes to increase coverage of the search space and help prevent premature convergence on a local optimum. Chromosomes are then evaluated according to a fitness (or objective) function, with the fittest surviving and the less fit being eliminated. The result is a gene pool that evolves over time to produce better and better solutions to a problem (Ragsdale, 2001). The GA's search process typically continues until a pre-specified fitness value is reached, a set amount of computing time passes, or until no significant improvement occurs in the population for a given number of iterations.

The key to find a good solution using a GA lies in developing a good chromosome representation of solutions to the problem. A good GA chromosome design should reduce or eliminate redundant chromosomes from the population. Redundancy refers to a solution being able to be represented by a chromosome in more than one way and appearing in the population multiple times. Multiple representations of the same solution increase the search space and slow the search.

A well-designed chromosome should reduce or eliminate redundancy, accurately represent a solution to the problem, and allow the GA operators to work effectively to generate better solutions as the iterative evolutionary process continues. An example of a chromosome representation designed for the efficient solution of a specific problem is found in the grouping GA developed by Falkenauer (1998). Falkenauer's grouping GA increases the effectiveness of GAs on grouping problems (i.e., bin packing, workshop layout and graph coloring) by modeling the problem in a way that reduces the redundancy in the population. A well-designed chromosome must also allow its fitness to be calculated easily. This facilitates comparison of the various chromosomes to determine which are better and should remain in the population.

Solving the TSP using GAs has generated a great deal of research on how best to perform the action of "evolving" an optimal (or good) solution to the problem. A number of different crossover methods have been proposed in the literature to solve the TSP using a GA. Some of the most commonly used operators include: Order Crossover (Goldberg, 1989; Davis, 1985; Starkweather et al., 1991; Oliver et al., 1987), Partially Mapped Crossover (Goldberg, 1989; Starkweather et al., 1991; Goldberg and Lingle, 1985; Oliver et al., 1987), Cycle Crossover (Goldberg, 1989; Starkweather et al., 1991; Oliver et al., 1987) and Asexual Crossover (Chatterjee et al., 1996; Schmitt and Amini, 1998; Fox and McMahon, 1991). Other proposed TSP operators include utilizing a matrix chromosome representation (Poon and Carter, 1995; Qu and Sun, 1999), hybrid operators (Liaw, 2000), simple crossover (Knosala and Wal, 2001), ordered crossover #2 (Syswerda, 1989) and moon crossover (Moon et al., 2002).

4. Chromosome representations for the MTSP

As mentioned earlier, two different chromosomes are commonly employed when solving the MTSP using GAs. We review these chromosomes below, discuss their properties and weaknesses, and then introduce a new chromosome that offers several advantages for solving the MTSP.

4.1. The one chromosome technique

Fig. 1 illustrates a method for representing solutions to a MTSP (where $n = 15$ and $m = 4$). This

Cities

2	5	14	6	-1	1	11	8	13	-2	4	10	3	-3	12	15	9	7

Salesperson 1 **Salesperson 2** **Salesperson 3** **Salesperson 4**

Fig. 1. Example of the one chromosome representation for a 15 city MTSP with four salespersons.

technique involves using a single chromosome of length $n + m - 1$ and is referred to as the "one chromosome" technique (Tang et al., 2000). In this technique, the n cities are represented by a permutation of the integers from 1 to n. This permutation is partitioned into m sub-tours by the insertion of $m - 1$ negative integers (from -1 to $-(m - 1)$) that represent the change from one salesperson to the next. Any permutation of these $n + m - 1$ integers represents a possible solution to the problem.

In the example in Fig. 1, the first salesperson would visit cities 2, 5, 14 and 6 (in that order). The second salesperson would visit cities 1, 11, 8 and 13 (in that order), and so on for salespersons 3 and 4. Using this chromosome design, there are $(n + m - 1)!$ possible solutions to the problem. However, many of the possible chromosomes are redundant. For example, simply exchanging the values of the first five genes with those of the next five genes would produce an equivalent (redundant) solution. Additionally, using this technique it is possible for two or more of the negative integers to appear consecutively—effectively reducing the number of salespersons to be utilized.

4.2. The two chromosome technique

Fig. 2 illustrates another approach for representing solutions to the same MTSP (where $n = 15$ and $m = 4$) that we refer to as the "two chromosome" technique. As the name implies, this method requires two chromosomes, each of length n, to represent a solution. The first chromosome provides a permutation of the n cities while the second assigns a salesperson to each of the cities in the corresponding position of the first chromosome (Malmborg, 1996; Park, 2001).

For example, in Fig. 2 cities 2, 8, 12 and 9 (in that order) are visited by salesperson 2. Cities 5, 14, 10 and 15 (in that order) are visited by salesperson 1; and likewise for the remaining cities being assigned to salespersons 3 and 4. Using this technique, there are $n!m^n$ possible solutions to the problem where n is the number of cities and m is the number of salespersons. However, many of the possible solutions are redundant. For example, the first two genes in each of the above chromosomes can be interchanged to create different chromosomes that result in an identical (or redundant) solution.

4.3. The two-part chromosome technique

Fig. 3 illustrates a new chromosome design for the same MTSP (with $n = 15$ and $m = 4$) that has two distinct parts; hence the name "two-part chromosome". The idea of using a two-part chromosome for the MTSP is similar to the two-part chromosome of the grouping GA (Falkenauer, 1998). The first part of the chromosome is a permutation of integers from 1 to n, representing

Cities

2	5	14	6	1	11	8	13	4	10	3	12	15	9	7

Salespersons

2	1	1	3	4	3	2	4	4	1	3	2	1	2	3

Fig. 2. Example of the two chromosome representation for a 15 city MTSP with four salespersons.

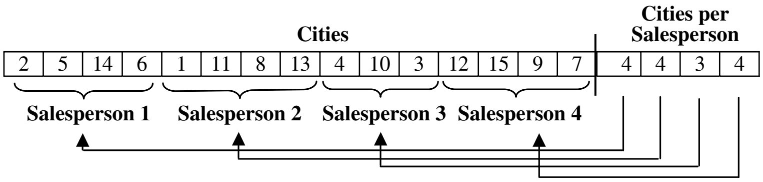


Fig. 3. Example of the two-part chromosome representation for a 15 city MTSP with four salespersons.

the n cities. The second part of the chromosome is of length m and represents the number of cities assigned to each of the m salespersons. The values assigned to the second part of the chromosome are constrained to be m positive integers that must sum to the number of cities to be visited (n).

In Fig. 3, salesperson 1 visits cities 2, 5, 14 and 6 (in that order), salesperson 2 visits cities 1, 11, 8 and 13 (in that order), and so on for salespersons 3 and 4. Using the new two-part chromosome method reduces the size of the search space due to the elimination of some (but not all) redundant solutions. For example, cities assigned to salesperson 1 always appear first in the two-part chromosome followed by the cities assigned to the second salesperson. This was not the case in either of the previous two chromosomes, where cities for a given salesperson could appear in any relative position in the chromosome (accounting for much of the redundancy in those chromosomes).

Using the two-part chromosome for the MTSP, there are $n!$ possible permutations for the first part of the chromosome. The second part of the chromosome represents a positive vector of integers $(x_1, x_2, \ldots, x_m)$ that must sum to n. There are $\binom{n-1}{m-1}$ distinct positive integer-valued m-vectors that satisfy this requirement (Ross, 1984). Thus, the solution space for the two-part chromosome is of size $n!\binom{n-1}{m-1}$.

5. Comparison of solution spaces

The three equations that describe the size of the solution space for each of the above chromosomes are summarized below. In Appendix A, we present proofs showing that the solution space for the two-part chromosome is smaller than those of the other two techniques when $n > m \geqslant 1$.

Technique	Size of solution space
One chromosome	$(n + m - 1)!$
Two chromosome	$n!m^n$
Two-part chromosome	$n!\binom{n-1}{m-1}$

Fig. 4 illustrates the magnitude of the differences in the solution spaces for the three chromosomes for a MTSP with $n = 15$ cities as the number of salespersons is varied from 1 to 14. Note that the y-axis on this chart represents the natural logarithm of the size of the solution space. When there is one salesperson ($m = 1$), the chromosomes default to a single TSP with identical solution spaces (i.e., $n! = 15! = 1.30767E{+}12$). As the number of salespersons (m) increases, the solution spaces of the traditional chromosomes quickly dwarf that of the new two-part chromosome. Of course, there is some extra computa-

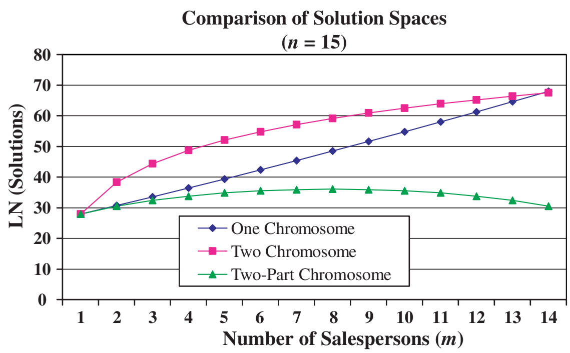


Fig. 4. Comparison of chromosome solution spaces for a 15 city MTSP with m salespersons.

tional overhead associated with ensuring that the second portion of the two-part chromosome satisfies the constraint imposed on it. However, our computational results (presented below) suggest this overhead is worth the investment, particularly for large (more difficult) MTSPs.

6. Computational testing methodology

In order to evaluate the practical benefits of the proposed two-part chromosome, computational tests were conducted to compare the performance of all three techniques on a set of problems created for the MTSP. The test problems were Euclidean, two-dimensional symmetric problems with 51, 100, and 150 cities. We refer to these problems as MTSP-51, MTSP-100, and MTSP-150, respectively. Table 1 summarizes the experimental conditions of 12 different problem size (n) and salesperson (m) combinations along with the run times allowed for each type of problem. For each level of n, all salespersons started and ended their individual tours in the same city.

Two different fitness (or objective) functions were considered. The first fitness function minimized the total distance traveled by all of the salespersons combined. For these runs, each salesperson was required to visit at least one city (other than the home city). This objective would represent a situation where there is a set number of salespersons and there are no constraints associated with the maximum number of cities visited by any one salesperson. A second set of runs was conducted using a fitness function that minimized the longest route among the m salespersons. This objective function would be used when scheduling to equalize the workload among the available salespersons.

Table 1
Computational test conditions

Number of cites (n)	Number of salespersons (m)	Run time (minutes)
51	3, 5, and 10	5
100	3, 5, 10, and 20	10
150	3, 5, 10, 20, and 30	15

Each of 12 problem size (n) and salesperson (m) combinations were run 10 times for each of the two different fitness objectives. The tests were run on a Pentium M, 1.7 MHz machine with 1024 MB of RAM using programs written in Visual Basic 6.0.

6.1. GA issues

The GA programs all utilized the ordered crossover method, roulette wheel selection, a population size of 100, and a 5% mutation probability. Ordered crossover was selected as it is a commonly used TSP operator. The values for population size and mutation probability also represent commonly used values. A detailed explanation of the operation of roulette wheel selection and the ordered crossover operator can be found in Goldberg (1989).

The second part of the two-part chromosome used a single point asexual crossover method (Chatterjee et al., 1996). This method simply cut the second part of the chromosome into two sections and reversed the order in which the two pieces were arranged. This type of crossover ensures that the second part of the chromosome remained feasible (with the sum of the values in the chromosome equaling n). Also, when the GA created a new child chromosome using this technique, the first part of the new child chromosome was matched with each of the second parts from the existing population as well as with the newly created second part. The fitness was determined for each possible pairing. If any of these pairings produced a better fitness than the current best solution, then the new pairing replaced the current best solution. If the new child solution was not better than the current best but was better than the parent, then the child solution replaced the existing parent chromosome.

The creation of the initial population for each of the three solution techniques was handled differently due to the varying structure of the chromosomes. The initial population generation strategies are summarized below.

6.1.1. One chromosome technique

Each chromosome is generated by randomly generating a permutation of the n cities along with

$m - 1$ negative integers to represent the change from one salesperson to another.

6.1.2. Two chromosome technique

The first chromosome is a randomly generated permutation of the n cities. The second chromosome is created by filling each position with a uniformly distributed random integer between one and the number of salespersons (m).

6.1.3. Two-part chromosome technique

The first part of each chromosome is a randomly generated permutation of the n cities. Recall that the sum of the m positive integers in the second part of the chromosome must equal n. For the tests using the fitness function representing the total distance of all salespersons, m gene values (x_i) for the second part of the chromosome were generated as a discrete uniform random number between 1 and $n - \sum_{k=1}^{i-1} x_k$, for $i = 1$ to m (i.e., the maximum value of each successive gene value was based on n and the sum of the previous values). These values were then randomly assigned to the genes in the second part of the chromosome. For the tests using the fitness function that minimized the longest of the m routes, n discrete uniform random numbers from 1 to m were generated and counted in their corresponding positions in the second part of the chromosome (i.e., $x_i = x_i + 1$ when the value i was generated).

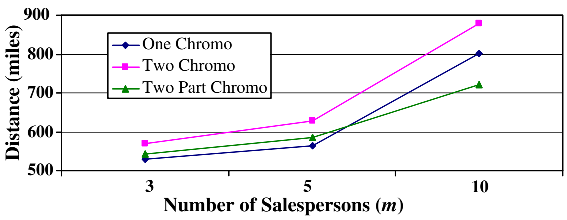
In addition to the solutions generated by the above, each of the starting populations was seeded with a solution produced by a simple greedy heuristic in order to give the GA a good starting point. The greedy solutions for the "minimize the total distance" problems were generated by looking at the present location of all the salespeople and finding the unassigned city that is the closest to one of the salespeople. The closest unassigned city is then assigned to the closest salesperson. This process was continued until all the cities were assigned. The greedy solutions for the "minimize the longest tour" problems were determined by rotating through all the salespeople in a round-robin fashion, assigning the closest unassigned city to each salesperson in turn, and continuing on until all the cities are assigned.

7. Computational results: minimum total distance

The first objective function sought to minimize the total distance of all the salespersons added together. This objective reflects the goal of minimizing the distance required to visit all n cities. The only constraint used with this objective was that each salesperson must visit one city (other than their home city). Without this constraint, the GA could have reduced the number of salespersons in the problem, thus possibly reducing the MTSP to a TSP (if all but one salesperson is assigned no cities).

When minimizing the total distance of all the salespersons in a MTSP, as the number of salespersons increases, the total distance of all of the trips also tends to increase. This results from the fact that each salesperson must start and return to the home city. So as the number of salespersons increases, the number of trips into and out of the home city increases, thereby increasing the total distance traveled. The results for the runs using the total distance fitness function are presented in Figs. 5–7.

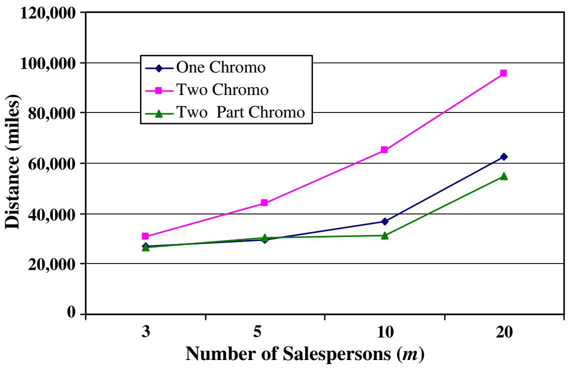
For the 51 city problem (MTSP-51), with the goal of minimizing total distance, the best chromosome representation varies depending on the number of salespersons (m). At the lower levels ($m = 3$ and 5), the one chromosome and two-part chromosome produced statistically equivalent results and the two chromosome performed significantly worse. With $m = 10$ salespersons, the two-part



	$m = 3$	$m = 5$	$m = 10$
One Chromosome	529	564	801[a]
Two Chromosome	570[a]	627[a]	879[a]
Two-Part Chromosome	543	586	723

[a] Indicates statistically significant differences from the two-part chromosome at the 0.01 level.

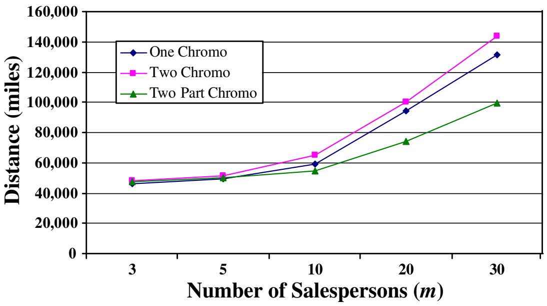
Fig. 5. Average miles traveled (51 city problem minimizing total distance).



	$m = 3$	$m = 5$	$m = 10$	$m = 20$
One Chromosome	27,036	29,753	36,890	62,471[a]
Two Chromosome	30,972[a]	44,062[a]	65,116[a]	95,568[a]
Two-Part Chromosome	26,653	30,408	31,227	54,700

[a] Indicates statistically significant differences from the two-part chromosome at the 0.01 level.

Fig. 6. Averages miles traveled (100 city problem minimizing total distance).



	$m = 3$	$m = 5$	$m = 10$	$m = 20$	$m = 30$
One Chromosome	46,111	49,443	59,341[a]	94,291[a]	131,503[a]
Two Chromosome	48,108	51,101	64,893[a]	100,037[a]	143,476[a]
Two-Part Chromosome	47,418	49,947	54,958	73,934	99,547

[a] Indicates statistically significant differences from the two-part chromosome at the 0.01 level.

Fig. 7. Averages miles traveled (150 city problem minimizing total distance).

chromosome produced superior results to the one chromosome representation. In all cases, the two chromosome technique produced significantly worse results relative to the other techniques.

For the 100 city problem (MTSP-100) with the goal of minimizing total distance, the best chromosome representation again varies depending on the number of salespersons (m) used. At the three lower levels of m ($m = 3$, 5 and 10), the one chromosome and two part representations produced statistically equivalent solutions. When the number of salespersons increased to $m = 20$, the two-part chromosome representation produced results that were statistically significantly better than the one chromosome representation. Here again, the two chromosome representation continued to produce the worst results.

For the 150 city problem (MTSP-150) with the goal of minimizing total distance, the best chromosome representation again varies depending on the value of m. At the two lower levels of m ($m = 3$ and 5), all three representations produced statistically equivalent solutions. For the higher levels of m tested ($m = 10$, 20 and 30), the two-part chromosome produced statistically significantly better results than the one chromosome and the two chromosome representations.

The results from the tests using total distance as the fitness function do not show a superior technique for all cases. For each of the different problems, the one chromosome and two chromosome representations produced results that were not statistically significantly different with a small number of salespersons. However, as the number of salespersons increased (and the solution space grew), the two-part method began to exhibit superior performance. In almost every case, the two-part method dominated the results of the two chromosome method. With the largest problem ($n = 150$ cities), the trend for the two-part method to surpass the one chromosome method was more dramatic than for the smaller problems. These results strongly suggest that as problem complexity increases, the proposed two-part chromosome is the superior solution alternative.

8. Computational results: minimum longest tour

While the objective of minimizing the total distance traveled is interesting, most real world MTSP applications are probably more interested in minimizing the longest individual salesperson's tour. Minimizing the longest tour has the goal of balancing the cities (or workload) among the salespersons and minimizing the distance the salespersons travel. For example, trucking companies might use this objective when creating a model to equalize the work load and minimize overtime hours (where the objective might involve hours versus miles). Balancing the loads is accomplished via the GA trying to move cities from the longest tour to other tours. At the same time the GA is also trying to shorten the length of the tours by changing the order the cities are visited within the tour.

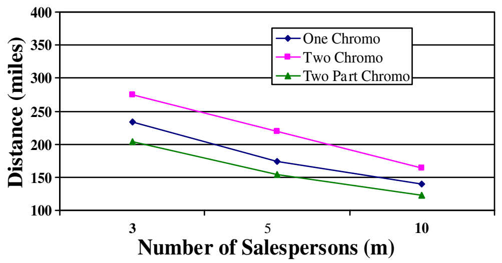
With the objective of minimizing the longest tour, the fitness values decrease as the number of salespersons increase because we are dividing the cities up among more salespersons. This creates a situation where each salesperson visits fewer cities, so the fitness values decrease (improve) with the addition of more salespersons.

We repeated our tests using the objective of minimizing the longest individual salesperson's tour. The results of these tests are presented in Figs. 8–10.

For the 51 city problem (MTSP-51) with the objective function of minimizing the longest tour, the two-part chromosome representation produced a statistically superior result at every level of m. The other two representations produced performed in a consistent pattern, with the one part representation performing better at every level of m then the two chromosome representation.

For the 100 city problem (MTSP-100) with the objective function of minimizing the longest tour, the two-part chromosome representation again produced a statistically significantly better result at every level of m. The other two representations produced performed consistently, with the one part representation performing better at every level of m then the two chromosome representation.

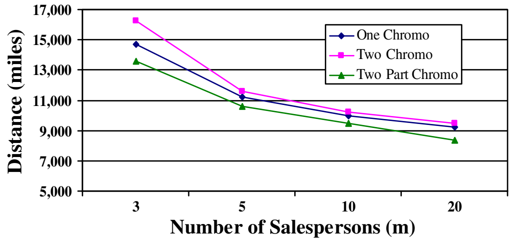
For the 150 city problem (MTSP-150) with the objective function of minimizing the longest tour, the two-part chromosome representation produced statistically superior results in most cases. The two-part chromosome always produced a superior result to two chromosome representation and when compared to the one chromosome representation the two part representation produced



	$m = 3$	$m = 5$	$m = 10$
One Chromosome	234[a]	173[a]	140[a]
Two Chromosome	275[a]	220[a]	165[a]
Two-Part Chromosome	203	164	123

[a] Indicates statistically significant differences from the two-part chromosome at the 0.001 level.

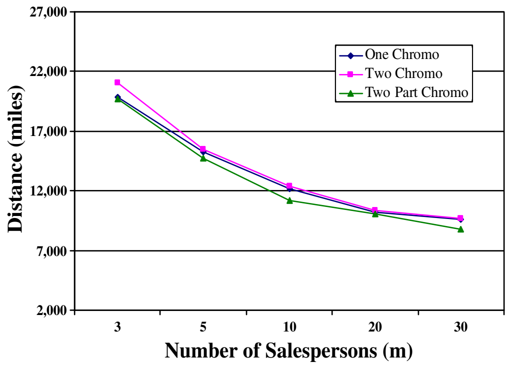
Fig. 8. Average longest tour (in miles) (51 city problem minimizing longest tour).



	$m = 3$	$m = 5$	$m = 10$	$m = 20$
One Chromosome	14,722[a]	11,193[a]	9,960[a]	9,235[a]
Two Chromosome	16,229[a]	11,606[a]	10,200[a]	9,470[a]
Two-Part Chromosome	13,556	10,589	9,463	8,388

[a] Indicates statistically significant differences from the two-part chromosome at the 0.005 level.

Fig. 9. Average longest tour (in miles) (100 city problem minimizing longest tour).



	$m = 3$	$m = 5$	$m = 10$	$m = 20$	$m = 30$
One Chromosome	19,875	15,229[a]	12,154[a]	10,206	9,626[a]
Two Chromosome	21,067[a]	15,450[a]	12,382[a]	10,338[a]	9,683[a]
Two-Part Chromosome	19,687	14,748	11,158	10,044	8,775

[a] Indicates statistically significant differences from the two-part chromosome at the 0.01 level.

Fig. 10. Average longest tour (in miles) (150 city problem minimizing longest tour).

statistically superior results at the $m = 5$, 10 and 30 levels. As with the 51 and 100 city problems, the two chromosome technique performed worse than either the one chromosome or two-part chromosome at every level of m.

The results of the testing conducted pursuing the objective of minimizing the longest tour clearly show the two-part technique to be the superior method. The two-part chromosome method found a statically significantly lower (better) fitness value than either of the other two methods in 10 out of the 12 tests run with the other two being statistically equivalent. The smaller solution space associated with the two-part chromosome clearly provides an advantage when searching for a balanced distribution of the workload among the salespersons. This can be of particular interest in areas such as production scheduling (balancing the workload amongst the available production lines) and in the VSP (balancing the load amongst the available trucks).

9. Conclusions

Modeling the MTSP using the new two-part chromosome proposed in this paper has clear advantages over using either of the existing one chromosome or the two chromosome methods. The two-part chromosome's smaller solution space allows it to produce solutions with better fitness values in most cases tested in this research (and all of the most difficult cases). When attempting to minimize the total distance traveled, the two-part chromosome was superior when the number of salespersons grew sufficiently large. When pursuing the "balanced workload" objective of minimizing the longest tour, the two-part chromosome produced superior results in 10 out of the 12 cases tested.

The two-part chromosome proposed in this paper has a number of areas of usage that will be of interest for future research. The two-part chromosome has performed well in theoretical and empirical comparisons to existing chromosomes for the MTSP. Additional research opportunities exist in the area of VSP, taking into account the various constraints and objectives the VSP faces. A variety of production scheduling problems may also benefit from the application of the new two-part chromosome introduced in this research. Finally, various perturbation strategies exploiting the two-part chromosome could be identified and tested to improve the performance of this modeling paradigm.

Appendix A

Here, we offer proofs showing that the solution space of our new, two-part chromosome is smaller than those of the other two chromosomes summarized again below.

Technique	Size of solution space
One chromosome	$(n+m-1)!$
Two chromosome	$n!m^n$
Two-part chromosome	$n!\binom{n-1}{m-1}$

We begin by demonstrating (via Theorem 1) that the solution space of the one chromosome technique exceeds that of the two-part chromosome whenever $n > m \geqslant 1$.

Lemma 1 (Ross, 1984, p. 14). *There are* $\binom{n-1}{m-1}$ *distinct positive integer-valued vectors* $(x_1, x_2, \ldots, x_m)$ *satisfying* $x_1 + x_2 + \cdots + x_m = n$ *where* $x_i > 0$, $i = 1, 2, \ldots, m$.

Lemma 2 (Ross, 1984, p. 14). *There are* $\binom{n+m-1}{n}$ *distinct nonnegative integer-valued vectors* $(x_1, x_2, \ldots, x_m)$ *satisfying* $x_1 + x_2 + \cdots + x_m = n$ *where* $x_i \geqslant 0$, $i = 1, 2, \ldots, m$.

Lemma 3. *For integers* $n > m \geqslant 1$, $\binom{n+m-1}{n} \geqslant \binom{n-1}{m-1}$. *(This follows immediately from Lemmas* 1 *and* 2.*)*

Theorem 1. *For integers* $n > m \geqslant 1$, $(n+m-1)! \geqslant n!\binom{n-1}{m-1}$.

Proof. From Lemma 3, observe that $n!\binom{n+m-1}{n} \geqslant n!\binom{n-1}{m-1}$.

Thus, it suffices to show that $(n+m-1)! \geqslant n!\binom{n+m-1}{n}$.

By contradiction, suppose $(n+m-1)! < n!\binom{n+m-1}{n}$.

Then, $(n+m-1)! < n!\binom{n+m-1}{n} = \frac{(n+m-1)!}{(m-1)!}$, which implies, $1 < \frac{1}{(m-1)!}$ for all $m \geqslant 1$.

However, $1 \geqslant \frac{1}{(m-1)!}$ for all $m \geqslant 1$.

Thus, $(n+m-1)! \geqslant n!\binom{n+m-1}{n} \geqslant n!\binom{n-1}{m-1}$. $\square$

We now show (via Theorem 2) that the solution space of the two chromosome technique exceeds that of the two-part chromosome whenever $n > m \geqslant 1$.

Lemma 4. *Given positive values A, B, C, and D, if $A \geqslant C$ and $B \geqslant D$ then $AB \geqslant CD$.*

Proof. By contradiction, if $AB < CD$ then $B/D < C/A$, but $B/D \geqslant 1$ and $C/A \leqslant 1$. $\square$

Theorem 2. *For integers $n > m \geqslant 1$, $n!m^n \geqslant n!\binom{n-1}{m-1}$.*

Proof. It suffices to show that $m^n \geqslant \binom{n-1}{m-1}$ for integers $n > m \geqslant 1$.

Let $n = m + k$ and assume $m^{m+k} \geqslant \binom{m+k-1}{m-1}$.

Observe that $m(k+1) \geqslant (m+k)$ or $m \geqslant (m+k)/(k+1)$ for all $m \geqslant 1$ and $k \geqslant 1$.

Thus, by Lemma 4,

$$mm^{m+k} \geqslant \frac{(m+k)}{(k+1)}\binom{m+k-1}{m-1} = \frac{(m+k)!}{(m-1)!(k+1)!}.$$

Hence,

$$m^{m+k+1} \geqslant \binom{m+k}{m-1}.$$

So, if Theorem 2 holds for any $n = m + k$, it holds for $n = m + k + 1$ and, by induction, any $n > m$.

Now, if $n = m + 1$, observe that $m^{m+1} \geqslant \binom{m}{m-1} = \frac{m!}{(m-1)!1!} = m$ for all $m \geqslant 1$.

Thus, Theorem 2 holds for all integers such that $n > m \geqslant 1$. $\square$

References

Bellmore, M., Malone, J., 1971. Pathology of traveling-salesman subtour elimination algorithms. Operations Research 19 (2), 278–307.

Carter, A.E., Ragsdale, C.T., 2002. Scheduling pre-printed newspaper advertising inserts using genetic algorithms. Omega 30 (6), 415–421.

Chatterjee, S., Carrera, C., Lynch, L., 1996. Genetic algorithms and traveling salesman problems. European Journal of Operational Research 93 (3), 490–510.

Davis, L., 1985. Job shop scheduling with genetic algorithms. In: Proceedings of the International Conference on Genetic Algorithms, London, pp. 136–140.

Falkenauer, E., 1998. Genetic algorithms and grouping problems. John Wiley & Sons, New York.

Fox, B.R., McMahon, M.B., 1991. Genetic operators for sequencing problems. In: Rawlings, G.J.E. (Ed.), Foundations of genetic algorithms. Morgan Kaugmann Publishers, San Mateo, CA, pp. 284–300.

Glover, F., 1990. Artificial intelligence, heuristic frameworks and tabu search. Managerial and Decision Economics 11 (5), 365–378.

Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading, MA.

Goldberg, D.E., Lingle, R.J., 1985. Alleles, loci, and the traveling salesman problem. In: Proceedings of the International Conference on Genetic Algorithms, London, pp. 154–159.

Holland, J.H., 1975. Adaptation in Natural and Artificial Systems. MIT Press, Cambridge, MA.

Jog, P., Suh, J.Y., Van Gucht, D., 1989. The effects of population size, heuristic crossover and local improvement on a genetic algorithm for the traveling salesman problem. In: Proceedings of the Third International Conference on Genetic Algorithms, Los Altos, CA, pp. 110–115.

Katayama, K., Sakamoto, H., Narihisa, H., 2000. The efficiency of hybrid mutation genetic algorithm for the traveling salesman problem. Mathematical and Computer Modeling 31 (10–12), 197–203.

Knosala, R., Wal, T., 2001. A production scheduling problem using genetic algorithm. Journal of Materials Processing Technology 109 (1–2), 90–95.

Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A., Shimoys, D., 1985. The Traveling Salesman Problem. John Wiley & Sons, New York.

Liaw, C., 2000. A hybrid genetic algorithm for the open shop scheduling problem. European Journal of Operational Research 124 (1), 28–42.

Little, J., Murty, D., Sweeney, D., Karel, C., 1963. An algorithm for the traveling salesman problem. Operations Research 11 (6), 972–989.

Malmborg, C., 1996. A genetic algorithm for service level based vehicle scheduling. European Journal of Operational Research 93 (1), 121–134.

Moon, C., Kim, J., Choi, G., Seo, Y., 2002. An efficient genetic algorithm for the traveling salesman problem with precedence constraints. European Journal of Operational Research 140 (3), 606–617.

Oliver, I., Smith, D., Holland, J., 1987. A study of permutation crossover operators on the traveling salesman problem. In: Proceedings of the Second International Conference on Genetic Algorithms, London, pp. 224–230.

Park, Y.B., 2001. A hybrid genetic algorithm for the vehicle scheduling problem with due times and time deadlines. International Journal of Productions Economics 73 (2), 175–188.

Poon, P.W., Carter, J.N., 1995. Genetic algorithm crossover operators for ordering applications. Computers and Operations Research 22 (1), 135–147.

Potvin, J., 1996. Genetic algorithms for the traveling salesman problems. Annals of Operations Research 63, 330–370.

Qu, L., Sun, R., 1999. A synergetic approach to genetic algorithms for solving traveling salesman problem. Information Sciences 117 (3–4), 267–283.

Ragsdale, C.T., 2001. Spreadsheet modeling and decision analysis, third ed. South-Western College Publishing, Cincinnati, OH.

Ross, S., 1984. Introduction to Probability Models. Macmillian, New York, NY.

Schaffer, J.D., Eshelman, L.J., Offutt, D., 1991. Spurious correlations and premature convergence in genetic algorithms. In: Rawlings, G.J.E. (Ed.), Foundations of genetic algorithms. Morgan Kaugmann Publishers, San Mateo, CA, pp. 102–112.

Schmitt, L., Amini, M., 1998. Performance characteristics of alternative genetic algorithmic approaches to the traveling salesman problem using path representation: An empirical study. European Journal of Operational Research 108 (3), 551–570.

Shirrish, B., Nigel, J., Kabuka, M.R., 1993. A Boolean neural network approach for the traveling salesman problem. IEEE Transactions on Computers 42 (10), 1271–1278.

Starkweather, T., Whitley, D., Whitley, C., Mathial, K., 1991. A comparison of genetic sequencing operators. In: Proceedings of the Fourth International Conference on Genetic Algorithms, Los Altos, CA, pp. 69–76.

Syswerda, G., 1989. Uniform crossover in genetic algorithms. In: Proceedings of the Third International Conference on Genetic Algorithms, Los Altos, pp. 502–508.

Tang, L., Liu, J., Rong, A., Yang, Z., 2000. A multiple traveling salesman problem model for hot rolling schedule in Shanghai Baoshan Iron and Steel Complex. European Journal of Operational Research 124 (2), 267–282.

Whitley, D., Starkweather, T., Fuquay, D., 1989. Scheduling problems and traveling salesmen: the genetic edge recombination operator. In: Proceedings of the Third International Conference on Genetic Algorithms, Los Altos, CA, pp. 133–140.

Wong, R., 1980. Integer programming formulations of the traveling salesman problem. In: Proceedings of the IEEE International Conference of Circuits and Computers, pp. 149–152.

Arthur E. Carter is an Assistant Professor in the school of Information Science and Technology at Radford University. He received his B.S. in Mechanical Engineering from Old Dominion University and holds an M.B.A. and Ph.D. in business information technology from Virginia Polytechnic Institute and State University. Dr. Carter's primary research interests are in the areas of artificial intelligence, optimization and decision support systems. He is a member of Decision Sciences Institute, the Institute for Operation Research and the Management Sciences and the Association for Information Systems.

Cliff T. Ragsdale is Bank of America Professor of Business Information Technology in the Pamplin College of Business at Virginia Tech. He received his Ph.D. in Management Science and Information Technology from the University of Georgia. He also holds an M.B.A. in Finance and B.A. in Psychology from the University of Central Florida. Dr. Ragsdale's primary area of research interest center on the integration of computers, mathematics, and artificial intelligence to solve business problems. He is a member of INFORMS, AIS and DSI. He has published in a variety of journals including *Decision Sciences*, *Decision Support Systems*, *Naval Research Logistics*, and *OMEGA*. He also serves on the Advisory Boards of *INFORMS Transactions on Education* and the *International Journal of Information Technology and Decision Making*. He is also author of the textbook *Spreadsheet Modeling and Decision Analysis* published by South-Western.